

Capstone Project Package (Final)

Pet Satisfaction Survey (UW-ITA343)

Anna van Raaphorst

March 17, 2006

Contents

- Overview
 - Project Overview
 - Project Definition
 - Key Requirements
 - Project Schedule
 - Application Architecture
- Requirements and Design Documents
 - Database Schema
 - Functional Requirements
 - Executive Change Orders (ECOs)
 - Code Design and Components
 - Form Variables
 - User Functions
- Project Report
 - Results (Features)
 - Lessons Learned
 - Release 2 Proposals

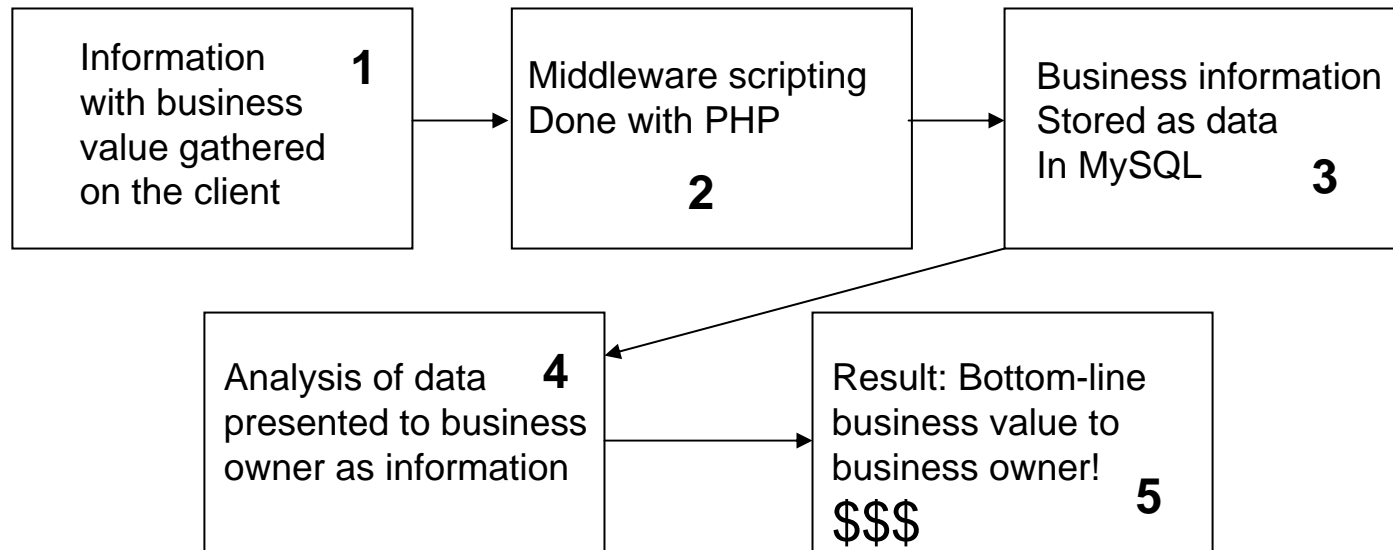
Overview

Project Overview

- Business driver: To increase the bottom-line profit of an e-commerce website selling pet-related products and services
- Online pet store owner wants to:
 - Collect profile data about website visitors for informational and marketing purposes
 - Encourage additional interest in the site and increase e-commerce purchases through a series of ballots rating “pets of the month”

Project Definition

Scripting language technology components and server-side database components integrated into a data-driven web application that has value to a business decision maker.



Key Requirements

- Project must simulate a typical project of its type from the Web developer's perspective
- Key components include:
 - A number of prototypes delivered throughout the course
 - Draft 1 of this document, including architecture and ERD drawings delivered in Lesson 3
 - Project package updated and approved in Lesson 4
 - Three major changes to the original design requiring Executive Change Orders (ECOs) in Lessons 5, 6, and 7
 - Final package (including Lessons Learned) delivered in Lesson 8

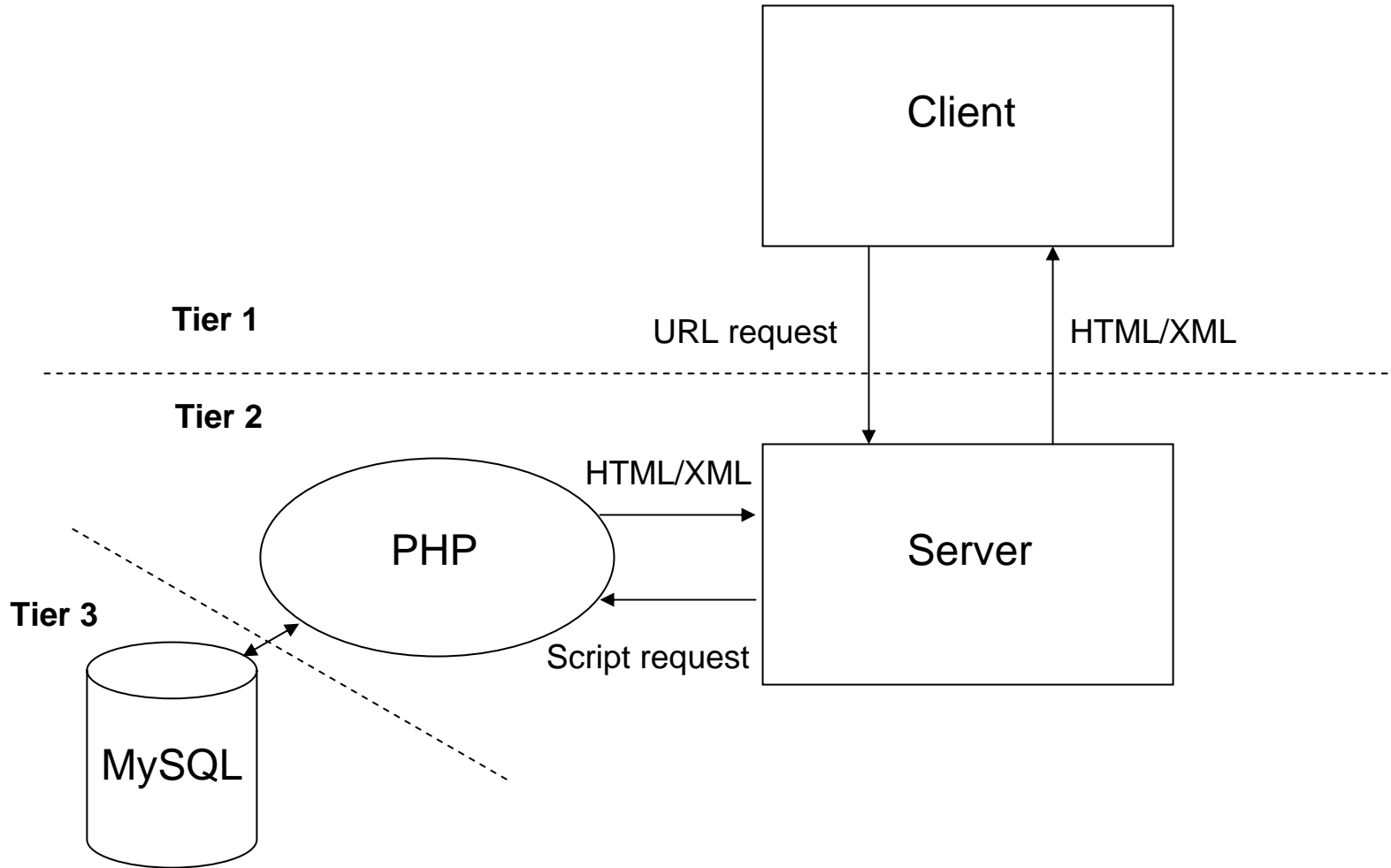
Key Requirements (continued)

- Risk will be mitigated by prototyping early in the project cycle and adjusting the requirements and projected final results throughout the cycle
- No data analysis is required for this project, but the data needs to be structured such that analysis could be done in a future phase of the project

Project Schedule

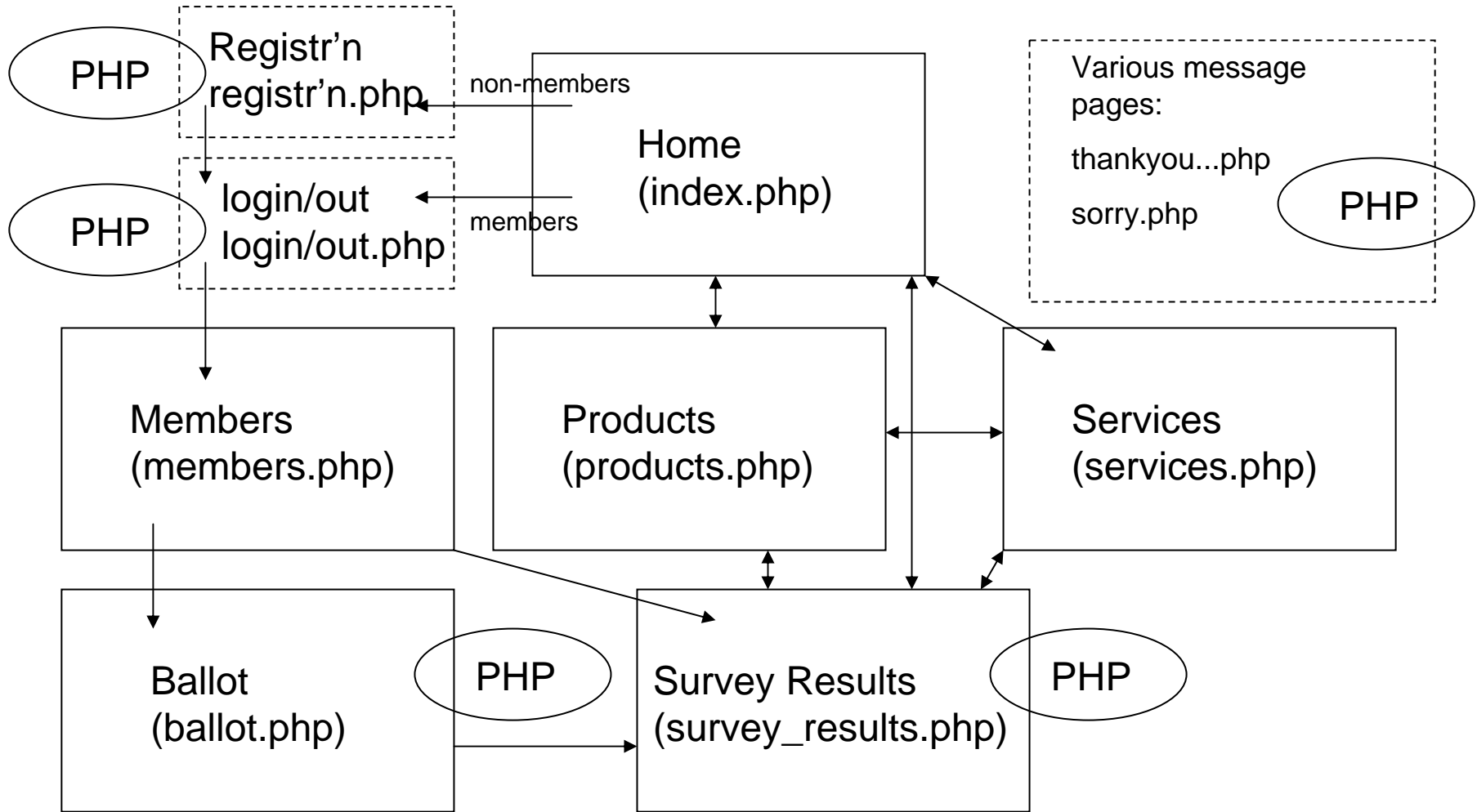
- February 27, 2006: Draft 1 of the project package (this document, with placeholders for some elements)
- March 6: Draft 2, after completion of Lesson 4 (actual: on time)
- March 13: Draft 3, after completion of Lesson 5 (actual: combined with Draft 4)
- March 20: Draft 4, after completion of Lesson 6 (actual: March 13)
- April 3: Final, after completion of Lesson 8 (actual: March 17)

Application Architecture

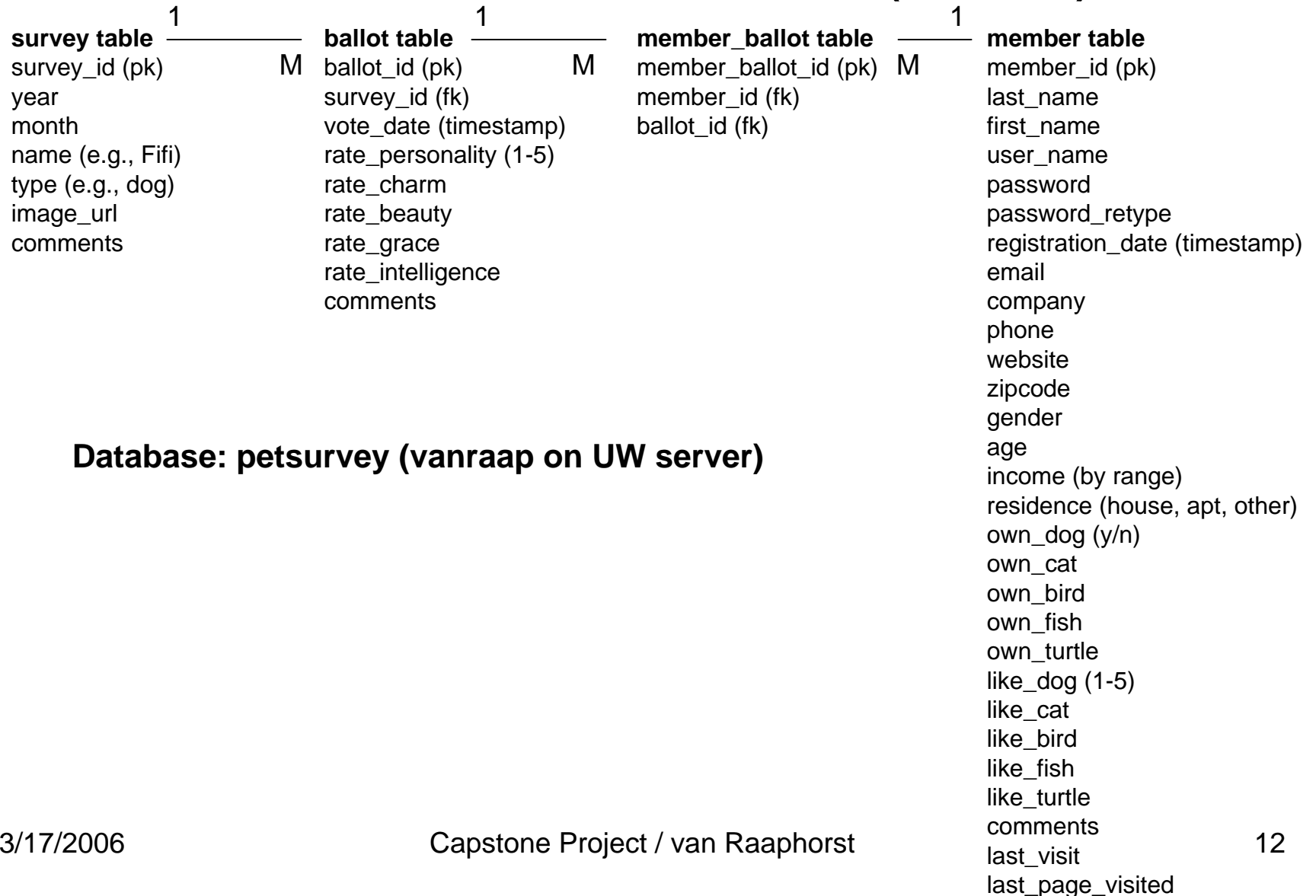


Requirements and Design Documents

Visible Pages



Database Schema (ERD)



Functional Requirements

- Casual visitors to the website will be encouraged to register as members, which entitles them to:
 - Discounts on pet products
 - Notification of product specials
 - Vote in the “pet of the month” ballots
 - A monthly newsletter about pets and pet products
- The membership questionnaire includes personal information (e.g., name, email address, income) pet ownership information, and general pet preferences
- This information will be used for marketing purposes
- Site visitors must be 18 or older to register as members (there is no enforcement code for this)

Functional Req'ts, continued

- The website will feature a monthly ballot in which members vote on the “pet of the month” (which could be a dog, cat, bird, fish, or turtle)
- Each ballot will include a photo of the pet and rating questions about its personality, charm, beauty, grace, and intelligence
- Ballot results will be posted on the site (in the general section, not only in the members-only section), and site visitors will be encouraged to participate in the new month’s ballot (of course, non-members must sign up first)

Functional Req'ts, continued

- A roll-up of current and prior ballots (collectively known as the “survey”) will be reported in monthly newsletters sent via email to members only
 - Newsletters themselves are not part of this project
 - However, the summary survey information is updated every time a ballot is processed, and kept as XML/RSS, so it would be readily available for inclusion in newsletters or other material
- Members could have the additional benefit of nominating their own pets for future ballots (possible Release 2 enhancement)

ECO #1 (Lesson 5)

- User authentication system to enable member registration (secure page)
- User information to be stored in the petsurvey database (member table)
- JavaScript validation of the form

ECO #2 (Lesson 6)

- Use cookies to store the encrypted password and the user name
- Use sessions to store first name, last name, email, and timestamp of last visit

ECO #3 (Lesson 7)

- Use XML to send snippets of data from the database to display in the browser as HTML – included in this release:
 - Summary (average score) of “pet of the month” ballot ratings for each month
 - Listing of comments for each month
- Use RSS to notify members when a new ballot is ready for their vote
 - RSS is produced in this release
 - Subscription/notification is a possible addition in a future release

Code Design and Components

- Site files visible to users:
 - index.php
 - login.php
 - logout.php
 - registration.php
 - members.php
 - ballot.php
 - survey_results.php
 - products.php (stub page, no direct relevance to this project)
 - services.php (stub page)
 - miscellaneous files like sorry.php and thankyou.php

Code Design, continued

- Include files:
 - connection.inc
 - header.inc
 - footer.inc
 - functions.inc
- CSS file:
 - UWCSS.css
- XML files:
 - rss.xml
 - sur_results1.xml
 - sur_results2.xml
 - sur_results3.xml

Form Variables

- Information to populate the member table (here are some examples, see the schema page for more information):
 - last_name
 - user_name
 - zipcode
 - own_dog
 - like_cat
 - etc
- Generally, the variable and column names are identical

Form Variables, continued

- Information to populate the ballot table (here are some examples, see the schema page for more information):
 - survey_id
 - vote_date
 - rate_charm
 - rate_intelligence
 - etc
- Generally, the variable and column names are identical

Key User Functions

- `escape_data()` for data validation
- `errmsg()` for handling error messages
- `issubset()` to test that all variables in the `$received` array are in the `$allowed` array
- `setlastpage()` to record the page the user was on when s/he logged out

JavaScript Program

- Validates the user_name and password input before they actually get submitted
- Located in the header file
- Produces an alert
- After this script runs, (additional) validation is done in PHP

Results (Key Features)

Usability Features

- Attractive, consistent web pages
- Extensible design
- Well-designed MySQL database
- Site-wide CSS file
- XHTML throughout
- Code is well laid out and thoroughly commented
- Common code in common files (e.g., connection.inc, header.inc, footer.inc, and functions.inc)
- Sticky forms in all forms
- Personalized greeting (first name) on login

Security Features

- Data validation using typecasting, JavaScript, and PHP (e.g., regular expressions, striptags)
- Data mismatch test to foil hackers – check that data received matches what is expected
- Encryption of password data using md5

Error-Handling

- User function `errmsg()`
- Clear, useful messages to user

Other Features

- Database already populated with meaningful sample data
- Database structured to facilitate meaningful business and marketing analysis
- Cookies and sessions used throughout

Lessons Learned

Prototyping

Description and Assessment

- I found it very useful to do the schema early, populate it with sample data, and use it to work out design bugs
- I made a number of schema changes based on the prototypes
- The project really came together for me when I created the entire website and walked through all the user actions using the prototyped pages and the database tables

Prototyping, continued

- I was glad to have the mybirthday script and table created in Lesson 3 of the course to give me a foundation for doing database inserts and updates using form data
- However, I found that my overall design for the Lesson 3 prototype needed to be reworked significantly
- I also reworked the design significantly after Lesson 4:
 - I had separated the visible pages (forms) from the handle_form pages
 - In Lesson 5 I pulled those pages back together

Prototyping, continued

- In Lesson 7 I created a `functions.inc` file as a place to keep all the functions
- Also in Lesson 7 I moved the script that makes the connection into `connection.inc`

Other Lessons Learned

- I need a PHP editor and IDE – I coded using notepad! I'm currently investigating some listed on the PHP editors webpage, or I will use Dreamweaver, which I use all the time for XHTML coding.
- I'm still having problems with the code logic among all the files (e.g., does a particular session variable need to be set before something else happens). I assume competence in this area comes only with lots of practice.

Release 2 Proposals

- Make it impossible for members to vote more than once for the pet of the month.
- Add a pet nomination scheme.
- Add administrative functionality (e.g., make it possible for the site admin to set a user password)
- Allow users to change their passwords
- Add lost password functionality

Release 2 Proposals, continued

- Add account (member) activation by email (click a link to activate)
- Tell users how to use the RSS capability to subscribe to the survey results file
- Analyze the data collected in the database